1-1-2010

Support-based distributed optimisation: an approach to radiotherapy patient scheduling

Graham Billiau
University of Wollongong, gdb339@uow.edu.au

Chee-Fon Chang
University of Wollongong, c03@uow.edu.au

Andrew Alexis Miller
University of Wollongong, amiller@uow.edu.au

Aditya K. Ghose
University of Wollongong, aditya@uow.edu.au

Recommended Citation
Billiau, Graham; Chang, Chee-Fon; Miller, Andrew Alexis; and Ghose, Aditya K.: Support-based distributed optimisation: an approach to radiotherapy patient scheduling 2010, 229-233.
https://ro.uow.edu.au/infopapers/1742

Support-based distributed optimisation: an approach to radiotherapy patient scheduling

Abstract

In the health system inefficiency leads to poor use of scarce expensive resources. Lengthy patient treatment waiting time can result from inefficiency in scheduling. The use of state-of-the art multi-agent and distributed computing technologies can provide a solution to address this problem. However, distributed optimisation in such a multi-agent setting poses an important challenge that requires protocols to enable agents to optimise shared objectives without necessarily revealing all of their private constraints. In this study we show that if the problem is expressed as a Dynamic Distributed Constraint Optimisation Problem a powerful algorithm such as SBDO can be deployed to solve it. As SBDO can be deployed in a grid all of the advantages of grid computing are also gained.

Keywords

Support, based, distributed, optimisation, approach, radiotherapy, patient, scheduling

Disciplines

Physical Sciences and Mathematics

Publication Details

Billiau, G., Chang, C., Miller, A. A. & Ghose, A. K. (2010). Support-based distributed optimisation: An approach to radiotherapy patient scheduling. In Healthgrid Applications and Core Technologies: Proceedings of HealthGrid 2010, 28-30 June, Orsay, France. Studies in Health Technology and Informatics, 159 229-233.

Support-Based Distributed Optimisation:

An Approach To Radiotherapy Patient Scheduling

Graham BILLIAU [a], Chee Fon CHANG [a,c], Andrew MILLER [b,c] and Aditya GHOSE [a]
[a] *Decision Systems Lab and Centre for Oncology Informatics, School of Computer Science and Software Engg, University of Wollongong, NSW, Australia, email:{gdb339,c03,aditya}@uow.edu.au*
[b] *Centre for Oncology Informatics, Illawarra Health and Medical Research Institute, University of Wollongong, NSW, Australia, email:alexisandrew@gmail.com*
[c] *Illawarra Cancer Care Centre, Wollongong Hospital, Wollongong NSW Australia*

Abstract. In the health system inefficiency leads to poor use of scarce expensive resources. Lengthy patient treatment waiting time can result from inefficiency in scheduling. The use of state-of-the art multi-agent and distributed computing technologies can provide a solution to address this problem. However, distributed optimisation in such a multi-agent setting poses an important challenge that requires protocols to enable agents to optimise shared objectives without necessarily revealing all of their private constraints. In this study we show that if the problem is expressed as a Dynamic Distributed Constraint Optimisation Problem a powerful algorithm such as SBDO can be deployed to solve it. As SBDO can be deployed in a grid all of the advantages of grid computing are also gained.

1. Introduction

The public health system is constantly under pressure to provide better services while working with limited resources. Minimising the patient treatment waiting time is a key problem area that requires attention. The obvious solution is to use the limited resources more efficiently. Patient scheduling is not a new problem. There exists commercial products such as [9] and studies such as [11] that attempt to address this issue. However, existing optimisation techniques such as linear programming and dynamic programming found in operations research are constrained by organisational structures and cannot be deployed efficiently in practise.

As described in [10] all the existing work uses multi-agent systems for distributed scheduling. However, most approaches to distributed scheduling such as [3,7] assume a homogeneous organisation and organisational units where there is a consensus on a scheduling technique used. This is because most approaches have been focused on scheduling patients within one unit. We believe that consideration should be made for scheduling problems across heterogeneous organisations and organisational units where scheduling techniques deployed in individual units are not universal and may be difficult to integrate. In these situations, the resulting locally generated solutions maybe difficult to integrate. Also, factors such as scalability and fault tolerance become more important. Furthermore, approaches specific to patient scheduling such as [7,11] only perform schedule improvement relying upon an initial schedule using a "first in, first served" approach. Then a the agents take turns attempting to improve the schedule in a round robin fashion. This greatly reduces the scalability of such systems. Studies such as [4] assume a static problem with unchanging constraints after modeling. In reality, we are commonly faced with a dynamic problem where the problem changes due to the discovery of new information even while a schedule is being generated. Finally, privacy is not considered when distributing scheduling across different organisations. It is conceivable that certain information related to an individual or organisation should remain private and not be disclosed to other parties. Given the highlighted issues, we believe this area warrants further investigation.

Our approach utilises the Support Based Distributed Optimisation (SBDO)[1] algorithm with the aim to provide a patient-centric deployment of agents technology in a distributed scheduling optimisation exercise similar to those found in [10]. However, we consider the distributed scheduling problem as a coordination problem rather than a decomposition of a larger problem into sub-parts. As such, we do not place restrictions on the heuristics or approaches deployed by individuals or organisations in generating solutions for their local scheduling problem. The only requirement is that the participants share a common communication language. As we consider it a coordination problem the same technology could be used to coordinate nodes in a grid infrastructure.

This allows us to view agents as nodes and shared constraints between the agents as arcs or relations facilitated by the communication channels. This makes it easy to deploy as a grid, with the agents for one unit hosted on a computer controlled by that unit, with the associated advantages of being scalable, dynamic and fault tolerant. A multi-agent approach is recapitulates the distribution of the problem between different units and provides established mechanisms for maintaining privacy. Finally the agents can easily be distributed within a grid computing framework with each agent being hosted in their unit. This approach is unique in providing a pareto-optimal solution balancing the goals of optimisation of resources and maximisation of patient satisfaction by meeting their preferences. Furthermore, the SBDO algorithm is completely asynchronous so it can easily scale to large problems.

In the remainder of the paper, we motivate our approach by presenting a description of the radiotherapy patient scheduling problem, present a formal framework and related algorithm, an encoding of the problem in our framework.

1.1. Problem Description

while scheduling staff and patients is widely applicable across the health system, for simplicity we restrict our discussion to just the treatment of cancer patients using radiotherapy.

We motivate our work with a description of the distributed patient scheduling problem. We will assume that there exists a collection of hospitals servicing a region. Each hospital operates independently such that they do not share any resources, and each contains a radiotherapy department. Each department also operates independently. In reality, while departments and hospitals can operate collaboratively to share resources, we will assume that no resources are shared.

Patients diagnosed with cancer have usually undertaken multiple procedures and examination. The diagnosed patient is scheduled to see an oncologist. The choice of which department to visit will depend on several factors including distance from the unit, waiting time to get an appointment for a specialist, waiting time before starting a treatment, the particular specialisation of the unit, referring doctor and patient preferences. Once the patient has seen a radiation oncologist, the patient is scheduled for a simulation where a patient-specific treatment plan is created by placing the patient on a simulator and performing a "dry run". The simulator machine is different to the radiotherapy treatment machine (a linear accelerator). Once simulation is completed and the treatment plan is generated, the patient is scheduled for treatment. In a course of radiotherapy treatment, a patient may require between 1 and 39 fractions. The required fractions depends on the type and stage of the cancer and is determined by the specialist usually before simulation. These fractions are usually delivered on consecutive business days.

Furthermore, certain cancers ("aggressive") may require compensatory treatment adjustments for public holidays where treatment is missed, or enforced delays to maintain minimum gaps between multiple daily treatment. Every day some time is reserved either to accommodate urgent patients or machine downtime. Once the patient has completed the course of radiotherapy treatment, the patient is then scheduled to see the specialist for follow-ups several weeks or months later within imaging prior to the appointments.

This small example demonstrates that patients require a mix of services from different departments and hospitals. A load-balancing exercise is required to minimise patient wait-time for each of these different services. This exercise should be performed while still satisfying the patient's preferences. Scheduling in a such an environment is highly dynamic. Finally each department is highly independent and will not tolerate external interference with their work-flow such as changing schedules or an externally generated solution that may be perceived to give other departments unfair advantage such as the normal practise of imposing a precomputed ordering does. This scheduling problem is a typical incarnation of a Dynamic Distributed Constraint Optimisation Problem (DynDCOP) with hard and soft constraints. DynDCOP is a not well studied variation of Distributed Constraint Optimisation Problem (DCOP). The distinguishing feature of DynDCOP is the fact that the problem being modelled and solved is dynamic. Therefore, the problem being addressed is changing over time as new information is discovered or current information becomes obsolete.

Although there exist algorithms such as such as ADOPT [6], DPOP [8] and NCBB [2] for solving a standard DCOP, for a variety of reasons such as their reliance on a predefined variable ordering these algorithms cannot be easily extended to solve DynDCOP. Apart from our proposed SBDO, the only existing DynDCOP algorithm is DynCOAA proposed by Mertens[5]. However, there exists several limitations of DynCOAA such as it is not fault tolerant and the requirement of global communication to update the pheromone trails. Support Based Distributed Optimisation (SBDO) is a DynDCOP solver that utilises a novel argumentation strategy that does not require variables/agents ordering specification or enforcement. This is of particular interest as placing an ordering on agents may skew the final solution.

In the next section, we will illustrate the modelling of the scheduling problem in SBDO. In section 3 we highlight how our approach works in a grid setting. In section 4, we present the conclusions

2. Radiotherapy Scheduling

Since SBDO can solve DynDCOPs, radiotherapy scheduling problems can be solved if encoded as a DynDCOP. Due to space constraints we shall describe the encoding informally. Each patient is represented by one agent. Only this agent knows the following private information about the patient: patient class (A1, A2, B1, B2, C, D or E), "aggressive" tumour, ready for care date, number of fractions (1–39), duration of each fraction in minutes, chemotherapy (none, concurrent or unsequenced), the patients preferences and, if receiving chemotherapy, the chemotherapy schedule. In order to represent the current schedule the agent has the following public variables: radiotherapy machine, fraction N date, fraction N start time, fraction N end time. So an agent can have between 4 and 118 public variables. Furthermore, in order to ensure a feasible schedule is generated, the following constraints must be enforced. Of these constraints only the first one is public, all the others are private:

- There must not be more than one patient scheduled on one machine at any time.
- The end time of each fraction must be 'duration' minutes after the start time of that fraction. Except for the first fraction, which takes 10 minutes longer.
- The first fraction must not be scheduled before the ready for care date.
- A patient must not receive two fractions within 6 hours of each other.
- Patients with "aggressive" tumours must receive 10 fractions per fortnight.
- Patients without "aggressive" tumours must receive 9 or 10 fractions per fortnight.
- Patients that are also receiving chemotherapy must have their radiotherapy schedule synchronised with their chemotherapy schedule.

Finally to optimise the schedule that is generated the following objectives are considered. All of these objectives are private:

- A patient should receive their treatment as soon as possible.
- Once a patient has been informed of their schedule each days treatment should not move by more than 15 minutes.
- The patients personal preferences should be respected.
- Patients without aggressive tumours should receive 10 fractions per fortnight.

Two other special agents are also required. The first manages downtime, whether public holiday or breakdown. Patients are not scheduled into public holidays. In the event of a breakdown this agent occupies the time slot, forcing the affected patients to reschedule. The second manages the time reserved for urgent patients and breakdown re-scheduling. It ensures that only patients with a high enough priority can be scheduled in these time slots

3. Grid Implications

When a new patient requires a schedule, a requester can pass the case to an agent in each radiotherapy department. The constraints of the particular case can be used to find the next available open appointments. Where a radiotherapy department's work flow has also been constrained to reflect its capacity, each department's agent can provide an estimate of the important scheduling which would affect decision making (i.e., date of oncologist appointment, date of simulation and date of treatment start). Although no private information need be released, the estimates returned to the requesting agent are sufficient to permit informed and automated decision making. Once the decision is made, scheduling can be commenced. Servicing the shortest delays will effectively load balance departments in close proximity.

4. Conclusion

The efficiency of hospitals can be significantly improved by the use of patient scheduling systems. We have outlined how Dynamic Distributed Constraint Optimisation, specifically our algorithm, SBDO, can be deployed in a grid computing infrastructure across an entire hospital or even an entire health system to manage all aspects of a patients treatment. This is a significant improvement from other solutions which can not scale. Further our approach does not require a homogeneous network. It will still work even when entities use different scheduling systems internally.

To illustrate our point we show how the small sub-problem of scheduling cancer patients for radiotherapy treatment can be modelled as a dynamic distributed constraint optimisation problem.

References

[1] Graham Billiau and Aditya Ghose. Sbdo: A new robust approach to dynamic distributed constraint optimisation. In Jung-Jin Yang, Makoto Yokoo, Takayuki Ito, Zhi Jin, and Paul Scerri, editors, *Principles of Practice in Multi-Agent Systems*, volume 5925 of *Lecture Notes in Computer Science*, pages 641–648. Springer, 2009.

[2] Anton Chechetka and Katia Sycara. No-commitment branch and bound search for distributed constraint optimization. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1427–1429, New York, NY, USA, 2006. ACM.

[3] G.Q. Huang, J.S.K Lau, K.L. Mak, and L. Liang. Distributed project scheduling with information sharing in supply chains: part 1 – an agent based negotiation model. *International Journal of Production Research*, 1:4813–4838, 2005.

[4] N. Liu, M.A. Abdelrahman, and S. Ramaswamy. A complete multiagent framework for robust and adaptable dynamic job scheduling. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 5:904–916, 2007.

[5] Koenraad Mertens. *An Ant-Based Approach for Solving Dynamic Constraint Optimization Problems*. PhD thesis, Katholieke Universiteit Leuven, Dec 2006.

[6] Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo. Adopt: asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161:149–180, 2005.

[7] Torsten O. Paulussen, Nicholas R. Jennings, Keith S. Decker, and Armin Heinzl. Distributed patient scheduling in hospitals. In *IJCAI*, pages 1224–1232, 2003.

[8] Adrian Petcu and Boi Faltings. Dpop: A scalable method for multiagent constraint optimization. In *IJCAI 05*, pages 266–271, Edinburgh, Scotland, Aug 2005.

[9] 2010. http://spectrasoft.com/practice-management-software/.

[10] Aysegul Toptal and Ihsan Sabuncuoglu. Distributed scheduling: a review of concepts and applications. *International Journal of Production Research*, 1:1–28, 2009.

[11] Ivan Vermeulen, Sander Bohte, Koye Somefun, and Han La Poturé. Multi-agent pareto appointment exchanging in hospital patient scheduling. pages 185–196, 2007.